

MODUL SISTEM PENGATURAN KECEPATAN MOTOR DC SECARA *REAL-TIME* BERBASIS LABVIEW

Muhammad Sulaiman, Zulftaman, Ermanu A.H.

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Muhammadiyah Malang

Kontak Person:

Muhammad Sulaiman

Jl. Raya Tlogomas 246 Malang, Telp. 0341-464318, Fax.0341-460782

E-mail: Muhsul0807@gmail.com

Abstrak

Proses pembelajaran yang menyajikan teori tanpa diikuti dengan praktek akan mempengaruhi kedalaman pemahaman yang diterima oleh mahasiswa. Seperti halnya dalam pembelajaran sistem kontrol, keberadaan sebuah modul modul pembelajaran yang dilengkapi Human Machine Interface (HMI) berbasis Labview yang terhubung langsung dengan plant yang akan dikontrol akan sangat bermakna. Tujuan dari studi ini adalah bagaimana merancang sebuah modul pembelajaran yang dapat digunakan dalam proses pembelajaran sistem kontrol yang mengkombinasikan penggunaan software, hardware dan perangkat interface. Modul ini memanfaatkan Labview sebagai HMI yang terhubung dengan motor DC sebagai plant yang dikontrol menggunakan NI USB 6008/6009. Teknik kontrol yang dirancang meliputi sistem kontrol PID, Fuzzy, Fuzzy-PID dan MRAC. Keempat jenis sistem kontrol tersebut digunakan untuk mengontrol kecepatan motor DC secara real-time. Aspek analisis yang dikembangkan pada modul ini adalah analisa transien keluaran sistem dan tingkat kemampuan sistem-sistem kontrol dalam mengkompensasi beban gangguan. Beban direpresentasikan dalam bentuk motor yang terhubung dengan motor DC. Hasil pengujian dengan beberapa metode kontrol. Dari proses pengujian diperoleh hasil bahwa modul pembelajaran yang dirancang dapat dipergunakan untuk memverifikasi rancangan beberapa sistem kontrol di atas secara real-time untuk beragam nilai pembebanan. Kemudian, sistem kontrol yang memiliki respon transien dan kemampuan mengkompensasi gangguan paling baik pada waktu settling time (t_s) = 28.169 detik pada kontrol MRAC, rise time (t_r) = 1.3618 detik pada kontrol FuzzyPID, nilai peak time (t_p) = 8 detik pada kontrol PID, dan nilai Overshoot (m_p) = 2.723 persen pada kontrol PID, sedangkan pada kontrol yang lain performanya cenderung lebih lama dari untuk mencapai kondisi tersebut. Dengan adanya modul pengajaran sistem kontrol ini diharapkan akan sangat membantu proses pembelajaran sistem kontrol baik secara praktis maupun analitik.

Kata kunci: modul pengajaran, sistem kontrol, Labview, kompensasi gangguan, real-time

1. Pendahuluan

Pada perkembangan dunia teknologi pemanfaatan motor Direct Current (DC) sangat banyak, baik dalam dunia pendidikan, dunia industri maupun bidang lainnya yang menggunakan motor DC sebagai aktuator atau sebagai penggerak. Motor DC merupakan motor listrik yang relatif mudah dalam pengontrolan maupun dalam pengendaliannya, peranan motor DC sangat penting yang mana dalam suatu proses penggunaannya membutuhkan tingkat keakuratan dan kepresisian yang tinggi karena hal tersebut sangat mempengaruhi hasil yang dikeluarkan. Dalam hal pengontrolan motor DC terdapat beberapa sistem maupun metode kontrol [1]. Diantara ialah kontrol Fuzzy, kontrol PID, ada pula penggabungan metode kontrol Fuzzy dan PID yang biasa disebut dengan kontrol Fuzzy PID [2].

Berdasarkan hal tersebut maka perlu ditingkatkannya kecakapan mahasiswa teknik elektro dalam menghadapi dunia industri, khususnya dalam bidang sistem kontrol. Adanya “Modul Human Machine Interface (HMI) Pengendalian Kecepatan Motor DC Menggunakan Metode Kontrol Bervariasi Berbasis LabView” diharapkan dapat menjadikan jawaban dari permasalahan diatas [3]. Pada studi ini penulis memanfaatkan HMI yang ada pada Labview sebagai media pembelajaran desain sistem kontrol yang akan diaplikasikan pada motor DC yang nantinya akan diberi gangguan dengan memutar motor DC yang dijadikan beban. Pada modul pembelajaran mahasiswa dapat melihat kemudian menganalisa perubahan respon beban ketika diberikan beberapa metode kontrol yang berbeda. Dengan demikian mahasiswa teknik elektro dapat memiliki kecakapan dalam penyelesaian masalah desain suatu sistem kontrol yang handal untuk sistem yang dinamis atau sistem yang memiliki gangguan yang variatif.

National Instruments LabView (NI Labview) adalah sebuah program yang berbentuk grafik terbuka yang ditetapkan oleh standar industri untuk aplikasi-aplikasi pengujian pengukuran dan otomasi. Sebuah perkembangan pada pemograman grafik di mana pengguna dapat

menggunakannya untuk mendesain suatu sistem menyerupai bentuk sistem aslinya, dapat melakukan pengamatan dengan hasil yang maksimal dan dapat mengontrol suatu aplikasi.

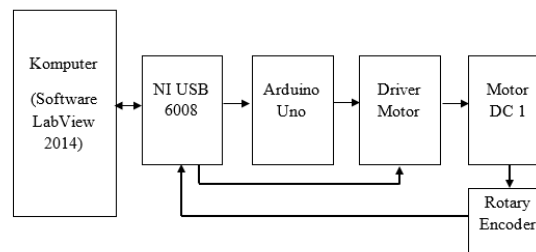
Tujuan dari studi ini adalah terciptanya sebuah modul pembelajaran sistem kontrol dengan memanfaatkan NI Labview sebagai HMI yang terintegrasi dengan NI USB 6008/6009 yang kemudian untuk mengontrol motor DC dengan menggunakan metode kontrol bervariasi agar mahasiswa dapat mengetahui dan menganalisa respon beban ketika menggunakan metode kontrol yang bervariasi.

2. Metode Penelitian

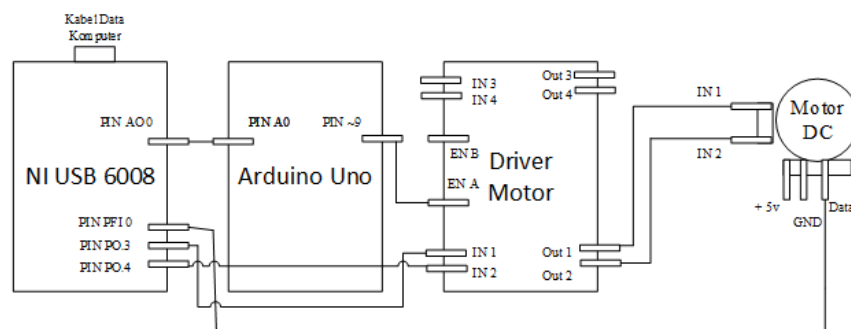
Pada bab ini menjelaskan tentang perancangan dan pembuatan sistem kontrol, baik secara *hardware* dan *software* yang akan digunakan untuk mendukung keseluruhan sistem yang akan diujikan. *Hardware* sendiri memiliki beberapa bagian yang saling terhubung dalam sistem, sedangkan *software* nanti akan di jadikan sebagai perantara antara komputer dengan *hardware*.

2.1 Diagram Blok Rangkaian Hardware

Pada rangkaian *hardware* terdiri dari beberapa komponen pendukung yang ditunjukkan dalam Gambar 1. Sebagai mana ditunjukkan dalam Gambar 1. komputer sebagai HMI antara *hardware* dan *software* NI Labview 2014 dimana komputer disini sangat berperan penting baik dalam pengontrolan maupun sebagai HMI dari program yang dibuat. NI USB 6008 disini digunakan sebagai media penerjemah dari komputer ke *hardware*, dimana nanti nya NI USB akan mengirim data ke Arduino melalui PIN NI USB AO 1 sebagai perintah untuk memberikan PWM ke *driver* motor, sedangkan hubungan NI USB 6008 dengan *driver* motor sebagai pemberi perintah untuk putar kanan maupun putar kiri pada *driver* motor dan hubungan dari ke komputer ketika ada perubahan data dari *rotary encoder*. Pada pembuatan modul pembelajaran sistem kontrol rangkaian pendukung *hardware* ditunjukkan dalam Gambar 2.



Gambar 1. Diagram blok prinsip kerja system

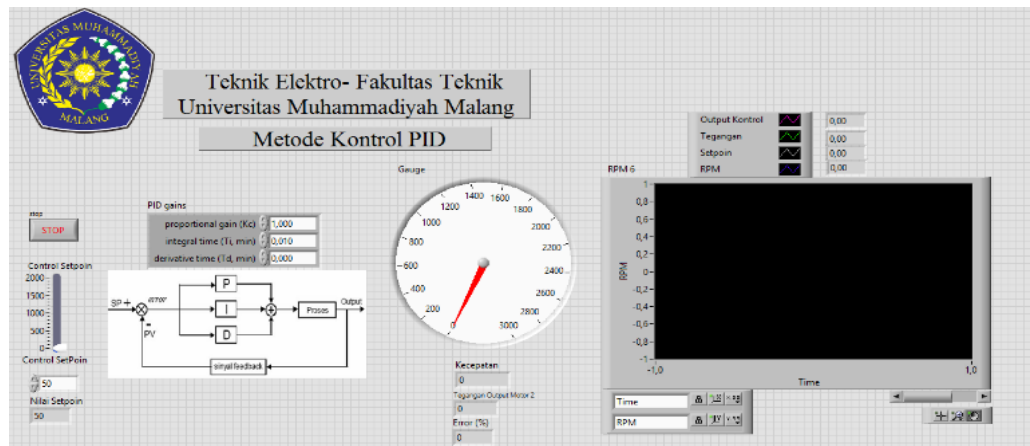


Gambar2. Rangkaian hardware

2.2 Perancangan Software

2.2.1 HMI Kontrol PID

Gambar 5 menunjukkan tampilan HMI dari kontrol PID yang mana terdiri dari PID system, nilai *setpoint*, *guage* kecepatan, beberapa indikator nilai, *Wave from graph* dan *button stop*. Dalam melakukan percobaan kita harus memasukkan nilai *setpoint*, *setting* nilai PID pada HMI lalu respon perubahan akan di tampilkan pada *Wave from graph*.



Gambar 5. HMI kontrol PID

Pada PID kontroler membandingkan *setpoint* (SP) ke *variabel proses* (PV) untuk mendapatkan kesalahan (e) yang digunakan pada persamaan 2. Untuk mewakili tindakan *proporsional*, *integral*, *derivative* dalam persamaan 3, 4, 5. Output Controller adalah penjumlahan dari *proporsional*, *integral*, dan aksi *derivatif*, seperti yang ditunjukkan dalam persamaan 6. Output kontroler sebenarnya terbatas pada kisaran tertentu untuk kontrol keluaran, pada persamaan 7 dan persamaan 8. Sedangkan untuk menunjukkan kontroler PID ditunjukkan dalam Persamaan 9.

$$e = SP - PV \quad (1)$$

$$u_p(t) = k_c e \quad (2)$$

$$u_I(t) = \frac{k_c}{T_i} \int_0^t e dt \quad (3)$$

$$u_D(t) = k_c T_d \frac{de}{dt} \quad (4)$$

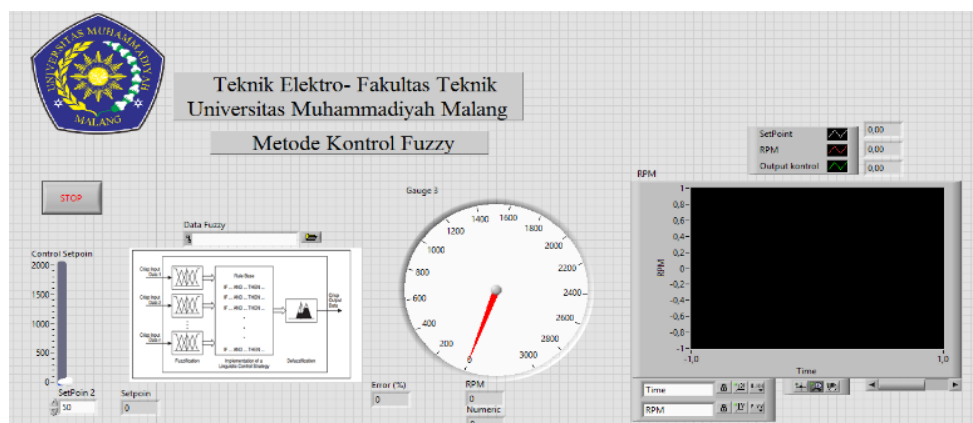
$$u(k) = u_p(k) + u_I(k) + u_D(k) \quad (5)$$

$$\text{if } u(k) \geq u_{\max} \text{ then } u(k) = u_{\max} \quad (6)$$

$$\text{if } u(k) \leq u_{\min} \text{ then } u(k) = u_{\min} \quad (7)$$

$$u(t) = k_c \left[(SP - PV) + \frac{1}{T_i} \int_0^t (SP - PV) dt - T_d \frac{dPV}{dt} \right] \quad (8)$$

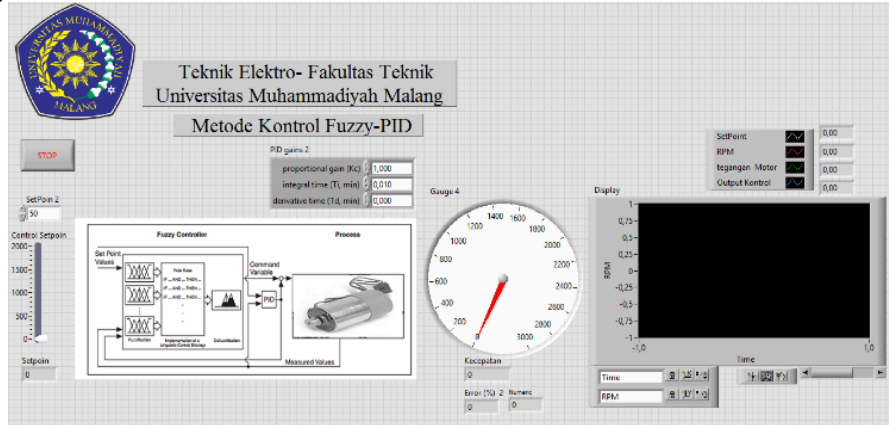
2.2.2 HMI Kontrol Fuzzy



Gambar 6. HMI Kontrol Fuzzy

Gambar 6 menunjukkan tampilan HMI dari kontrol Fuzzy yang mana terdiri dari Fuzzy sistem, nilai *setpoint*, guage kecepatan, beberapa indikator nilai, *Wave from graph* dan button stop. Dalam melakukan percobaan kita harus memasukkan nilai *setpoint*, *setting* nilai Fuzzy system pada HMI lalu respon perubahan akan di tampilkan pada *Wave from graph*.

2.2.3 HMI FuzzyPID



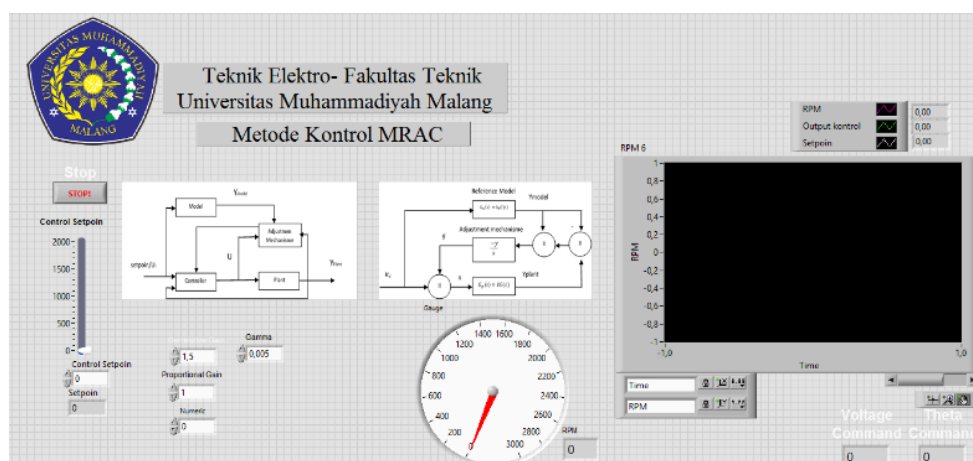
Gambar 7. Human Machine Interface (HMI) Kontrol FuzzyPID

Gambar 7 menunjukkan tampilan HMI dari kontrol Fuzzy-PID yang mana terdiri dari Fuzzy system, PID system, nilai *setpoint*, *guage* kecepatan, beberapa indikator nilai, *Wave from graph* dan *button stop*. Dalam melakukan percobaan kita harus memasukkan nilai *setpoint*, *setting* nilai PID pada HMI lalu respon perubahan akan di tampilkan pada *Wave from graph*.

kontrol FuzzyPID dapat dilihat pada nilai *setpoint* dan nilai *rotary encoder* dimasukan kedalam *toolkit* PID dan Fuzzy sistem yang mana di dalam nya telah diproses sebelum dikeluarkan pada *plant* motor DC. Pada aplikasi ini Fuzzy berfungsi menghitung *parameter* kontrol PID (K_p , T_i , T_d), berdasarkan kondisi sinyal *error* dan perubahan *error* dari *plant*.

Fuzzy dan PID kontroler bekerja di *paralel*. Proses menambahkan sinyal output dari kedua controller, tetapi sinyal output dari kontroler Fuzzy adalah nol di bawah operasi yang normal kondisi. PID kontroler output akan mendorong proses. Fuzzy kontroler mengintervensi hanya ketika mendeteksi kondisi operasi normal.

2.2.4 HMI Kontrol MRAC



Gambar 8. HMI kontrol MRAC

Gambar 8 menunjukkan tampilan HMI dari kontrol *Model Reference Adaptive Control* (MRAC) yang mana terdiri dari *System* MRAC, *Nilai setpoint*, *Guage* kecepatan, beberapa indikator nilai, *Wave from graph* dan *button Stop*. Yang mana ketika melakukan percobaan harus memasukkan

nilai *setpoint*, *setting* nilai Fuzzy system pada HMI dan respon perubahan akan di tampilkan pada *Wave from graph*. MRAC terdiri dari dua *loop*. *Loop* yang pertama adalah *loop* umpan balik pada umumnya. Sedangkan *loop* yang kedua adalah *loop* yang menuju pada blok *adjustment* parameter (*mekanisme* penyesuaian) untuk membandingkan kesalahan. Permasalahan dalam MRAC ini adalah menentukan *mekanisme* penyesuaian agar sistem menjadi stabil dan tidak ada kesalahan dalam keadaan tunak. *Mekanisme* perhitungan *parameter* kontroler pada *direct* MRAC terjadi dengan mempertimbangkan perbandingan *input* dan *output plant*. Pada *Listing program* kontrol MRAC dapat dilihat pada nilai *setpoint* dan nilai *rotary encoder* dimasukan kedalam system MRAC yang mana di dalam nya telah diproses sebelum dikeluarkan pada *plant* motor DC.

Dimana *Reference Model* ditunjukkan dalam persamaan 8, lalu persamaan *Controller* ditunjukkan dalam persamaan 9,10, Persamaan *adjustment* ditunjukkan dalam persamaan 11. Dimana *Theta 1* ditunjukkan dalam persamaan 12 dan juga *Theta 2* ditunjukkan dalam persamaan 13. Dan dimana persamaan *plant* ditunjukkan dalam persamaan 14, sehingga Jadi didapatkan persamaan 15 akhir *yplant*.

$$\frac{Y_m(s)}{U(s)} = \frac{b}{s^2 + a_1s + a_0} \quad (8)$$

$$K_p = e(t) \quad (9)$$

$$K_d = \frac{de(t)}{dt} \quad (10)$$

$$adjustment = \frac{-\gamma}{s} \quad (11)$$

$$Theta1(\theta_1) = \frac{a_1s + a_0}{s^2 + a_1s + a_0} \quad (12)$$

$$Theta2(\theta_2) = \frac{a_1s + a_0}{s^2 + a_1s + a_0} * u \quad (13)$$

$$plant = \frac{1.89}{s^2 + 0.0389s + 10.77 + 1.89} \quad (14)$$

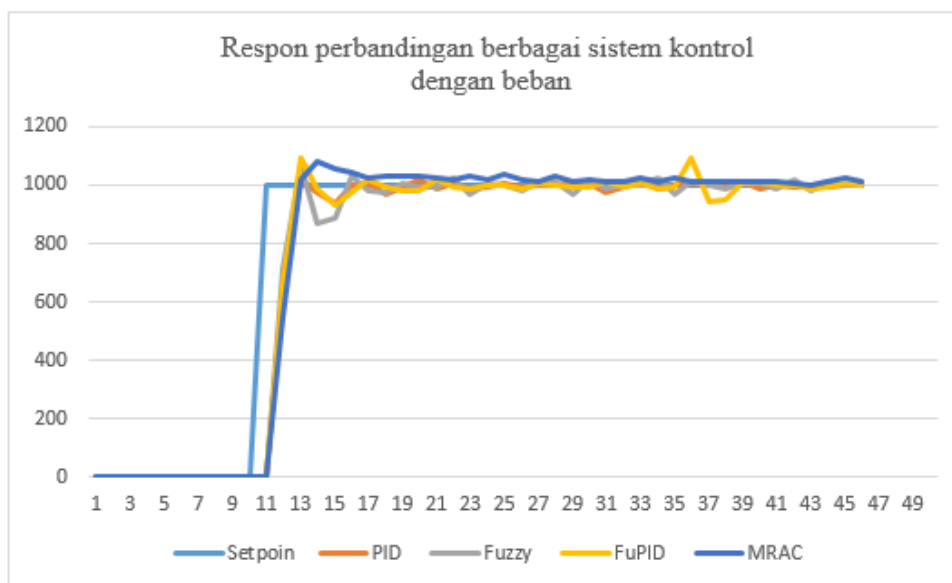
$$y_{plant} = \frac{1.89\theta_1}{s^2 + 0.0389s + 10.77 + 1.89\theta_2} u_c \quad (15)$$

3. Hasil dan Pembahasan

Pada bab ini akan dibahas tentang pengujian dan analisis berdasarkan perencanaan dari system yang dibuat. Program pengujian disimulasikan dalam HMI LabView dengan system yang telah di buat. Pengujian ini dilaksanakan untuk mengetahui kehandalan dari berbagai macam system kontrol yang telah dibuat dan untuk mengetahui hasil dari setiap kontrol yang di lihat dari grafik keluaran kontrol masing. Pengujian pertama-tama dilakukan secara per kontrol, dan kemudian dilakukan ke dalam system yang terintegrasi.

3.1 Perbandingan Sistem kontrol dengan beban

Gambar 9 dapat dilihat berbagai respon sistem kontrol dengan beban terhadap dengan perubahannya terhadap nilai *setpoint*, hingga nilai respon mencapai kondisi stabil. Pada pengujian untuk mengetahui respon dari berbagai kontrol nilai *setpoint* diatur dari 0 kemudian dinaikan pada nilai *setpoint* 1000. Ternyata dengan berbagai metode kontrol yang ada kita dapat membandingkan bagaimana respon tiap kontrol dari respon sinyalnya. Pemberian nilai kontrol pada sistem kontrol tergantung dari user yang memasukan nilai-nilai parameter kontrol untuk mendapatkan hasil sesuai dengan keinginan dari user, memasukan *parameter* dapat dilakukan melalui HMI yang telah di sediakan sebelumnya.



Gambar 9. Respon sistem kontrol dengan beban

3.2 Respon Sinyal dengan beban

Hasil data respon percobaan dari berbagai sistem kontrol dengan beban, dapat dilihat pada tabel 1. menjelaskan hasil pengujian berbagai sistem kontrol dengan beban untuk dapat dijelas sinyal step tiap kontrol antara lain : sinyal *rise time*, sinyal *overshoot*, sinyal *settling time*, *peak time*. Yang mana setiap respon sinyal dapat mengerti dan menyimpulkan dari semua sistem kontrol yang telah diujikan.

Tabel 1. Respon Sinyal tanpa beban

	PID	Fuzzy	FuzzyPID	MRAC
RiseTime(tr)	15.085	13.732	13.618	15.831
SettlingTime(ts)	481.187	481.431	492.318	281.693
SettlingMin	940.5	871.5	931.5	1002
SettlingMax	1018.5	1071	1096.5	1083
Overshoot(mp)	27.231	75.301	119.449	72.808
Peak:	1264	1071	1096.5	1083
PeakTime(tp)	8	13	13	14

4. Kesimpulan

Setelah melakukan pengujian terhadap modul pembelajaran sistem kontrol yang telah dibuat dan melakukan percobaan terhadap beberapa sistem kontrol dapat disimpulkan, yaitu:

1. Modul HMI sistem kontrol kecepatan motor DC menggunakan *software* aplikasi Labview 2014 dan hardware interface NI USB 6008/6009.
2. Pada modul pembelajaran sistem kontrol yang dibangun dan rancang, terdiri dari beberapa jenis metode kontrol antara lain : PID, Fuzzy, Fuzzy-PID dan MRAC.
3. Tampilan HMI yang telah dibangun dan dirancang ini akan mempermudah untuk menganalisa performa pada motor DC. Sehingga dapat mengetahui langsung respon ketika adanya perubahan setiap input kontrol yang diberikan.

4. Respon sinyal kontrol sistem berbeban menunjukan nilai terbaik *rise time* (t_r) = 1.3618 pada kontrol FuzzyPID, nilai *settling time* (t_s) = 28.1693 pada kontrol MRAC, nilai *peak time* (t_p) = 8 pada kontrol PID, dan nilai *overshoot* (mp) = 2.7231 pada kontrol PID, sedangkan pada kontrol yang lain performanya cenderung lebih lama dari untuk mencapai kondisi tersebut.

Daftar Notasi

SP	= Set Point (Nilai yang diinginkan)
PV(t)	= Present Value pada waktu=t (Nilai aktual)
Error(t)	= Error pada waktu=t
Error(t-1)	= Error pada waktu =t-1
Nilai s	= Gamma
γ	= Tuning parameter
a1	= 2,67
b dan a0	= 3,56
u	= nilai setpoint

Referensi

- [1] PETRELLA, Roberto, "Speed Measurement Algorithms for Low-Resolution Incremental Encoder Equipped Drives: a Comparative Analysis", University of Udine, Italy, 2008.
- [2] Zulfatman and M. F. Rahmat, "Application of self-tuning fuzzyPID controller on industrial hydraulic actuator using system identification approach", UTM Skudai, Malaysia, 2009.
- [3] PID and Fuzzy Logic Toolkit User Manual Labview 372192D-01, 2009.